CoinWindFinance Audit Report

Version 1.0.0

Serial No. 20211109000220 20

Presented by Fairyproof

November 9, 2021

FAIRYPROOF

01. Introduction

This document includes the results of the audit performed by the Fairyproof team on the CoinWind's single token staking and contract upgrade, at the request of the CoinWind team.

Audit Start Time:

November 6, 2021

Audit End Time:

November 10, 2021

Audited Source Files:

The calculated SHA-256 values for the audited files when the audit was done are as follows:

```
CICowToken.sol     :
0x2b4c9b16973c59332b5bcaf2868b94bc7e0f97544bcfc4dd87cd1a1f02844f6f
HubCommon.sol      :
0xcf169c5c73d153115d56187af5887cb474ccad914a6db8a7d2d629c439fcdef3
HubPool.sol        :
0x5cc150513f25651a9b3a972ea7ee78af4ad92d1215201a74f4f4106aec309289
HubPool2.sol       :
0x1fc80854a1d986fb880834500c92f9b3f53a57380ce7c0f0e984ff086f852bae
HubPoolExtend.sol  :
0xdaa352129e1f594b2097543cfcada54d31522ae669fdd93a21476fd5c2705fde
HubPoolProxy.sol   :
0x3dccbe3b9b14b3642d736ecc5f1e4f572a614e34f897e3d4ac5a323de98b1e62
HubPoolStorage.sol :
0x54f613555ecae1be893cf938445fd61d71687ee3e95a955b1f25587676b0eae4
MigratorProxy.sol  :
0x148fde3373a37c948ad9021ad6ef958bafc78c8018cadbbad930f924f237553e
MigratorStorage.sol:
0xea783f60bc182247e32c4ddad10c6b22b3d729293b6bfb8a51a8d1721453d959
MigratorV2.sol     :
0xa6e83c43581c5f8cd74f9a1514ffa70bd814956164f86de8920a005791f5bcf5
AddressSetLib.sol  :
0xd0da82ad1a71dac96388d6a1aeb10ab26105181cf59880498c99e40b406c7dbd
```

The source files audited include all the files with the extension "sol" as follows:

```
contracts/
├── CICowToken.sol
├── HubCommon.sol
```

```
├── HubPool.sol
├── HubPool2.sol
├── HubPoolExtend.sol
├── HubPoolProxy.sol
├── HubPoolStorage.sol
├── MigratorProxy.sol
├── MigratorStorage.sol
├── MigratorV2.sol
└── library
    └── AddressSetLib.sol

1 directory, 11 files
```

The goal of this audit is to review CoinWind's solidity implementation for its single coin staking and contract upgrade functions, study potential security vulnerabilities, its general design and architecture, and uncover bugs that could compromise the software in production.

We make observations on specific areas of the code that present concrete problems, as well as general observations that traverse the entire codebase horizontally, which could improve its quality as a whole.

This audit only applies to the specified code, software or any materials supplied by the CoinWind team for specified versions. Whenever the code, software, materials, settings, enviroment etc is changed, the comments of this audit will no longer apply.

— Disclaimer

Note that as of the date of publishing, the contents of this report reflect the current understanding of known security patterns and state of the art regarding system security. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk.

The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. If the audited source files are smart contract files, risks or issues introduced by using data feeds from offchain sources are not extended by this review either.

Given the size of the project, the findings detailed here are not to be considered exhaustive, and further testing and audit is recommended after the issues covered are fixed.

To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

— Methodology

The above files' code was studied in detail in order to acquire a clear impression of how the its specifications were implemented. The codebase was then subject to deep analysis and scrutiny, resulting in a series of observations. The problems and their potential solutions are discussed in this document and, whenever possible, we identify common sources for such problems and comment on them as well.

The Fairyproof auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Fairyproof to make sure we understand the size, scope, and functionality of the project's source code.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Fairyproof describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run the test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the source code to improve maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

— Structure of the document

This report contains a list of issues and comments on all the above source files. Each issue is assigned a severity level based on the potential impact of the issue and recommendations to fix it, if applicable. For ease of navigation, an index by topic and another by severity are both provided at the beginning of the report.

— Documentation

For this audit, we used the following sources of truth about how the single token staking and the upgraded contracts should work:

contract files

These were considered the specification, and when discrepancies arose with the actual code behavior, we consulted with the CoinWind team or reported an issue.

— Comments from Auditor

Serial Number	Auditee	Audit Time	Result
2021110900011018	Fairyproof Security Team	November 6, 2021 - November 10, 2021	Medium to Low Risk

Summary:

The Fairyproof security team used its auto analysis tools and manual work to audit the project. During the audit 2 vulnerabilities of medium-severity, 8 vulnerabilities of low-severity were found and 1 neutral suggestion was listed.

Among these vulnerabilities, 2 vulnerabilities of medium-severity and 5 vulnerabilities of low-severity were confirmed, 2 vulnerability of low-severity was fixed, 1 neutral suggestion was adopted, 1 vulnerability of low-severity was ignored.

02. About Fairyproof

Fairyproof is a leading technology firm in the blockchain industry, providing consulting and security audits for organizations. Fairyproof has developed industry security standards for designing and deploying blockchain applications.

03. Introduction to CoinWind

CoinWind is a smart yield farming DeFi application.

04. Major functions of audited code

The audited code was to upgrade the HubPool contracts by implementing a proxy mechanism such that the contracts could be upgraded/migrated and new functions could be added.

The audited contracts included:

`MigratorV2.sol`: an implementation contract which migrates the data and states of the HubPool contracts.

`HubPoolExtend.sol`: it adds interfaces for new functions.

`CICowToken.sol`: a certificate token contract.

`HubPool.sol` and `HubPool2.sol`: new implementation contracts for HubPool. **Note: staking assets to or withdrawing assets from new pools can be paused.**

`HubPoolProxy.sol` : the proxy contract for HubPool

`MigratorProxy.sol`: the proxy contract for Migrator

Note: the controller and strategy contracts were not covered by this audit

05. Coverage of issues

The issues that the Fairyproof team covered when conducting the audit include but are not limited to the following ones:

- Re-entrancy Attack
- Replay Attack
- Reordering Attack
- DDos Attack
- Transaction Ordering Attack
- Race Condition
- Access Control

- Integer Overflow/Underflow
- Timestamp Attack
- Gas Consumption
- Inappropriate Callback Function
- Function Visibility
- Implementation Vulnerability
- Uninitialized Storage Pointer
- Arithmetic Precision
- Tx.origin
- Fake Deposit
- Shadow Variable
- Design Vulnerability
- Token Issurance
- Admin Rights
- Inappropriate Proxy Design
- Inappropriate Use of Slots
- Asset Security
- Contract Upgrade/Migration
- Code Improvement

06. Severity level reference

Every issue in this report was assigned a severity level from the following:

Critical severity issues need to be fixed as soon as possible.

High severity issues will probably bring problems and should be fixed.

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Neutral is not an issue or risk but a suggestion for code improvement.

07. List of issues by severity

Index	Description	Issue	Severity	Status
N1	Unbounded Parameter Setting	Implementation Vulnerability	Low	Fixed
N2	Missing Validity Check for Pools	Implementation Vulnerability	Low	Fixed
N3	Unnecessary Rights	Code Improvement	Neutral	Fixed
N4	Able to Transfer Saved Assets	Admin Rights	Medium	Confirmed
N5	Excessive Number of Admins	Admin Rights	Medium	Confirmed
N6	Transfer of Certificate Tokens Can Be Paused	Admin Rights	Low	Confirmed
N7	Certificate Tokens Can Be Burned	Admin Rights	Low	Confirmed
N8	Core Parameters Can Be Reset	Design Vulnerability	Low	Confirmed
N9	Existing Tokens Could Be Repeatedly Added	Design Vulnerability	Low	Confirmed
N10	Contract Upgrade/Migration	Contract Upgrade/Migration	Low	Confirmed
N11	Missing Conditional Check for Contract Upgrade	Code Improvement	Low	Ignored

08. Issue descriptions

[N1] [Low] Unbounded Parameter Setting

Risk Severity: Low

Issue: Implementation Vulnerability

Description:

The `setPoolInfo` function in the `HubPoolExtend.sol` file didn't have a bounded setting for the value of `pool.feeRate` .

Recommendation: consider adding a bounded setting for the value of `pool.feeRate`

Status: the CoinWind team has fixed this.

[N2] [Low] Missing Validity Check for Pools

Risk Severity: Low

Issue: Implementation Vulnerability

Description:

The `updatePool` function in the `HubPool.sol` file didn't check whether or not the `pid` already existed. When a non-existing `pid` was input the function would still proceed and execute `pool.lastRewardBlock = block.number;`

Recommendation: consider adding a validity check for the pools.

Status: it has been fixed.

[N3] [Neutral] Unnecessary Rights

Risk Severity: Neutral

Issue: Code Improvement

Description:

The `HubCommon.sol` contract granted unnecessary rights to the `caller` thus making the caller have the same access control as the `controller`.

Recommendation: consider removing the unnecessary rights such as `checkCaller`.

Status: the CoinWind team has removed the unnecessary rights and the `caller` has different access control from the `controller`.

[N4] [Medium] Able to Transfer Saved Assets

Risk Severity: Medium

Issue: Admin Rights

Description:

The `inCaseTokensGetStuck` function in the 在`HubPool2.sol` file could be used to transfer the assets which were left in the contract and were not used in investment.

Recommendation: consider removing this function.

Status: the CoinWind team has confirmed this but prefers to keep it for now and plans to transfer the access control to this function to a multi-sig wallet in a future upgrade.

[N5] [Medium] Excessive Number of Admins

Risk Severity: Medium

Issue: Admin Rights

Description:

Both the HubPool and migrator contracts inherited the `AuthHub` contract. Therefore each of these contracts could be accessed by all of four roles `owner`, `governance`, `controller` and `caller`. In addition, each `CICowToken` contract could be accessed by both `owner` and `operator`. Each of these roles was granted to both some external accounts and contract accounts. These roles had previlleges such as token minting, setting the address of `controller`, and were able to set some core parameters. This resulted in excessive number of admins in existence and made it complicated to manage access control.

Recommendation: consider renouncing some accounts that had these previlleges and transferring some access control to multi-sig wallets.

Status: the CoinWind team has confirmed this and committed to managing the access control with great caution and care.

[N6] [Low] Transfer of Certificate Tokens Can Be Paused

Risk Severity: Low

Issue: Admin Rights

Description:

The `CICowToken.sol` contract distributed a certificate token to users. The certificate token was a standard ERC-20 token. The admin was able to pause its distribution.

Recommendation: consider removing this admin right.

Status: it has been confirmed by the CoinWind team but the team prefers to keep it for now.

[N7] [Low] Certificate Tokens Can Be Burned

Risk Severity: Low

Issue: Admin Rights

Description:

The `CICowToken.sol` contract distributed a certificate token to users. The certificate token was a standard ERC-20 token. The admin can burn this certificate token.

Recommendation: consider removing this admin right.

Status: it has been confirmed by the CoinWind team but the team prefers to keep it for now.

[N8] [Low] Core Parameters Can Be Reset

Risk Severity: Low

Issue: Design Vulnerability

Description:

The `setHubPoolV2` function and the `setHubPool` function in the `MigratorV2.sol` file, and the `setMigrator` function in the `HubPool2.sol` file could be arbitratily used to reset some core parameters which should only be set once.

Recommendation: consider re-implementing these functions such that the core parameters are only set once.

Status: it has been confirmed the CoinWind team but the team prefers to keep it for now.

[N9] [Low] Existing Tokens Could Be Repeatedly Added

Risk Severity: Low

Issue: Design Vulnerability

Description:

The `add` function in the `HubPool.sol` file had a comment saying identical tokens cannot be added repeatedly. However the function didn't implement this and relied on external inputs to avoid this issue.

Recommendation: consider defining a `mapping` variable to record all the added tokens and adding a validity check to prevent this issue.

Status: it has been confirmed by the CoinWind team but the team prefers to keep it for now.

[N10] [Low] Contract Upgrade/Migration

Risk Severity: Low

Issue: Contract Upgrade/Migration

Description:

The newly developed HubPool contract implemented a proxy mechanism that is easy for contract upgrade/migration.

Recommendation: consider transferring the access control to contract upgrade/migration to a multi-sig wallet and doing contract audits prior to contract upgrade/migration.

Status: the CoinWind team plans to transfer the access control to a multi-sig wallet after the new contracts are deployed.

[N11] [Low] Missing Conditional Check for Contract Upgrade

Risk Severity: Low

Issue: Code Improvement

Description:

The `HubPool` supported contract upgrade/migration but didn't have conditional checks for contract upgrade. When a contract upgrade/migration failed , unexpected issues or risks would happen.

Recommendation: consider adding a conditional check for contract upgrade/migration.

Status: the CoinWind team doesn't consider this as an issue and prefers to keep it for now.

09. Recommendations to enhance the overall security

We list some recommendations in this section. They are not mandatory but will enhance the overall security of the system if they are adopted.

- Consider implementing a simplyfied and minimized proxy contract which only contains necessary variables such as "owners" and "implements" and moving all the auxiliary variables to its implementation contracts.

Update: the CoinWind team prefers to keep it for now and plans to refine its implementation in the future.

- A token whose transaction amount will be deducted in a transfer transaction should not be allowed to be used as a staking asset or reward asset.

Update: the CoinWind team has acknowledged this and will not allow tokes of this kind to be used as staking or reward assets.